|  |  |
| --- | --- |
| **Node**  **Callback**   * Rappeler lorsque terminer * Pas besoin d’attendre après la tâche * Morceaux de code exécuté après   **Event-Loop**   * Personne n’attend, il y a une file d’exécution   **À l’examen**   * Un bout de code d’un service qui ne fonctionne pas (P-t rapport avec des sync)   **NPM**   * Gère les dépendances   + Se fit sur « package.json » pour travailler. * Gère la version | **Android**  **Fragment**   * User control en wpf * Utiliser des parties d’interface dans plrs écrans. Ils peuvent comm entre eux. On peut l’utiliser sur plsrs appareils Paysage / Portrait   **Navigation Drawer**   * Menu de navigation * Fonctionne avec les fragments * On remplace le contenu de l’écran avec des fragments * Pour que des fragments communiquent entre eux, l’activité qui les contient doit implémenter l’interface de communication : « Must be implement … » ; « On.Listener »   **Adapter**   * Manière d’afficher une collection dans l’écran. * Fai t le lien entre une ListView et les données |
| **Thread**   * C’est quoi : Démarrer plsrs tâches en parallèle. Ordre aléatoire (On peut forcer l’ordre en synchronisant plusieurs thread (Sémaphore / lock)) * Exercice : Set Time-out   **Type de service**   * SAAS (Software as a service)   Logiciel sans être installé sur le poste de l’utilisateur  • Hébergé sur un serveur du fournisseur, rendu via Internet • Gratuit ou abonnement • Accessible de partout (fureteur, client léger) • Installation, mise à jour fournie par le fournisseur • Non adapté pour logiciel spécifique à l’entreprise • Exemple : Google Apps, Microsoft Office 365   * IAAS (Infrastructure as a service)   + Machine virtuelle * STAAS (Storage as a service)   + Dropbox   + ICloud   + Mega   + Google Drive   + One Drive   + Réseaux virtuelle   + Pare-feu   + Balanceur de charge   + Ex : Amazone * PAAS (Platform as a service)   + On a accès à une machine virtuelle  pour développer, mais on n’a pas  accès à son matériel   + C9   **SOA (Service-oriented Architecture)**  https://www.akana.com/images/solutions/300x250px-_API-soa.png  **Cloud Computing**   * C’est quoi ? * Est-ce que vous connaissez des applications infonuagiques * Selon vous ça commencera quand   **Socket**   * Canal de communication généralement en temps réel * Sans socket = Pas internet (temps réel)   **HTTP Code**   * 200 : Ok * 501 : Server error * 204 : No content * 201 : Content Created * 404 : Not found * 418 : Im a Tea Pot   **Service Web VS Site Web**   * Site web -> Mr et Mme Tous le monde * Service web -> Pour les Développeur   **Serveur Web avec VS sans état**   * State Less: Le client est responsable de donner tout eels info nécéssaires au serveur.   State : Le serveur conserve les informations du client et ce dernier n’est pas responsible de ses info  **Web 2.0**  https://upload.wikimedia.org/wikipedia/commons/thumb/3/38/Carte_web_2.png/310px-Carte_web_2.png  **Web 3.0**   * Le web intelligent   + Plus seulement des ordis co   + Tout (Tel / Refrigi / Auto)   + Système multiagent et distribué   + Sématique * Composant à composant   + L’humain n’a plus à intervener | |